

noe

noe

noe

noe

MICROCOMPUTERS

MICROCOMPUTERS

MICROCOMPUTERS

MICROCOMPUTERS

FINALLY!

a microcomputer family that
delivers on all the buzzwords !!

- ✓ Lowest Cost
- ✓ Highest Performance
- ✓ Ease of Use
- ✓ Upward Expansion
- ✓ Best Addressing
- ✓ Single Supply
- ✓ Simple I/O
- ✓ Modular System
- ✓ Single Phase Clock

*Buy a microprocessor
for \$20.00 and see
for yourself.*

The MOS Approach

- - the third generation

The MCS6500 product line is a third generation microcomputer family, combining the best features of second generation families into a product line that is both a price and performance leader.

The application of advanced but proven fabrication techniques plus superior architectural design allow for incorporation of advanced addressing features (a limitation of all second generation microprocessors such as the MC6800, 8080 and F8) in chips of smaller size allowing for expanded performance at lowest cost.

A unique feature of the MCS6500 Product line is the MCS650X microprocessor family which will include a range of software compatible microprocessors with performance/cost compatibilities ranging from 8 bit microprocessors which are price competitive with the lowest cost 4 bit machines up to microprocessors which out-perform the best of today's 16 bit products.

To allow for minimum I/O cost and maximum user flexibility all of the MCS6500 products are compatible with the M6800 bus structure.

SYSTEM DESIGN

- Single 5 Volt Supply
- N-Channel, Silicon Gate, Depletion Load Technology
- Direct TTL Compatibility
- DMA Capability
- 65K Addressable bytes
- Total bus compatibility with the MC6800 family
- Full line of memory and support chips

COMPONENTS OF THE MICROCOMPUTER FAMILY

- **MCS6501** - Microprocessor Unit with External Two Phase Clock (40 pin package)
 - * Compatible with MC6800
 - * 65K addressable bytes of memory
- **MCS6502** - Microprocessor Unit with On-The-Chip Clock (40 pin package)
 - * External Single Phase Input
 - * RC Time Base Input
 - * Crystal Time Base Input
- **MCS6503** - Microprocessor Unit (28 pin package)
 - * Two interrupts
 - * 4K addressable bytes of memory
 - * On-The-Chip Clock
- **MCS6504** - Microprocessor Unit (28 pin package)
 - * One Interrupt
 - * 8K addressable bytes of memory
 - * On-The-Chip Clock
- **MCS6530** - A one chip combination
 - * RAM - 64 x 8
 - * ROM - 1024 x 8
 - * I/O - 16 Bi-directional pins
 - * Timing - Programmable intervals
- **MCS6102** - RAM (2102 type), 1024 x 1
 - * No hold time
 - * Faster access time
 - * Microprocessor compatible
- **MCS6111** - RAM (2111 type), 256 x 4
 - * No hold time
 - * Faster access time
 - * Microprocessor compatible

FUTURE PRODUCTS WILL INCLUDE

- **MCS6520** - Peripheral Interface Adapter
- **MCS6540** - 2048 x 8 ROM
- Additional Microprocessor options
- Numerous I/O combinations

SOFTWARE SUPPORT

- *Cross Assembler* - Allows programming of the microprocessor at symbolic assembly level. Provides extensive error checking and cross referencing information and is available on various time-sharing services, as well as available for purchase by the user.
- *Emulator* - Utilizing an input file generated from the Cross Assembler, the emulator determines the viability of operation and calculates the timing of sections of code. "Fortran like" control facilitates rapid verification of coding in either interactive or batch mode entry as well as time-sharing.

SYSTEM DEVELOPMENT SUPPORT

- *Microcomputer Development Terminal - MDT*
A high level development tool for modeling the system, the MDT allows the user to utilize a modular and flexible technique to verify the system design before committing to a finalized design. Interactive control allows the user to assemble programs, debug software through on-line editing capability and to program PROM's when the design is deemed correct. Interface to the MDT can be either through the included keyboard/display or via a provided port for use with teletype or higher speed peripherals. This unit contains a resident assembler to allow the user to engage this system as his only development tool. A standard option card allows for interactive debugging of pre-production and final production systems.
- *Teletype Input/Output Monitor - TIM*
An MCS6530 programmed to allow the user to input data directly into memory from teletype, to read memory via teletype, to load the output from the cross assembler into memory and to initiate programs from any memory location. Capability includes loading and writing from a high speed port.
- *Keyboard Input/Output Monitor - KIM*
An MCS6530 programmed to allow the user to interface to his own applications via a keyboard and alphanumeric display.

DOCUMENTATION

- *Hardware Manual*
Contains a detailed discussion of all chips in the family, how they interface, how the peripherals are controlled as well as design techniques to facilitate system operation, testing and maintenance. Specific emphasis is placed on "bringing up" a system, including testing techniques, scope synchronizing and general procedures used in trouble-shooting a system.
- *Programming Manual*
Defines the architecture of the MCS650X products. Defines each instruction and its effect on the internal registers with particular emphasis on the sophisticated addressing modes utilized in the MCS650X family. Contains detailed information on programming the MCS650X products.
- *Cross Assembler Manual*
Discusses the concept of assembler directives and the use of the assembler in time-share and batch operations. Particular emphasis is placed on understanding and resolving error messages.
- *Emulator Manual*
Discusses emulation techniques with emphasis on understanding errors and the process of resolving hardware and software problems using both the time-share and batch operations. This includes leading the user through the evolution of a sample program.
- *Systems Handbook*
A concise distillation of the major characteristics of the instruction set including operation codes, addressing modes and microprocessor status for each instruction. Also includes reference to the primary aspects of system implementation including chip interfaces and timing. This document is aimed at the experienced user who understands the basics of the family and requires a concise reference book.
- *MDT Manual*
Instructs the user in the operation of the MDT System and its application in developing a working microprocessor system.
- *TIM Manual*
Defines how to apply the Teletype Input/Output Monitor to developing microprocessor systems.
- *KIM Manual*
Defines how to apply the Keyboard Input/Output Monitor to developing microprocessor systems.



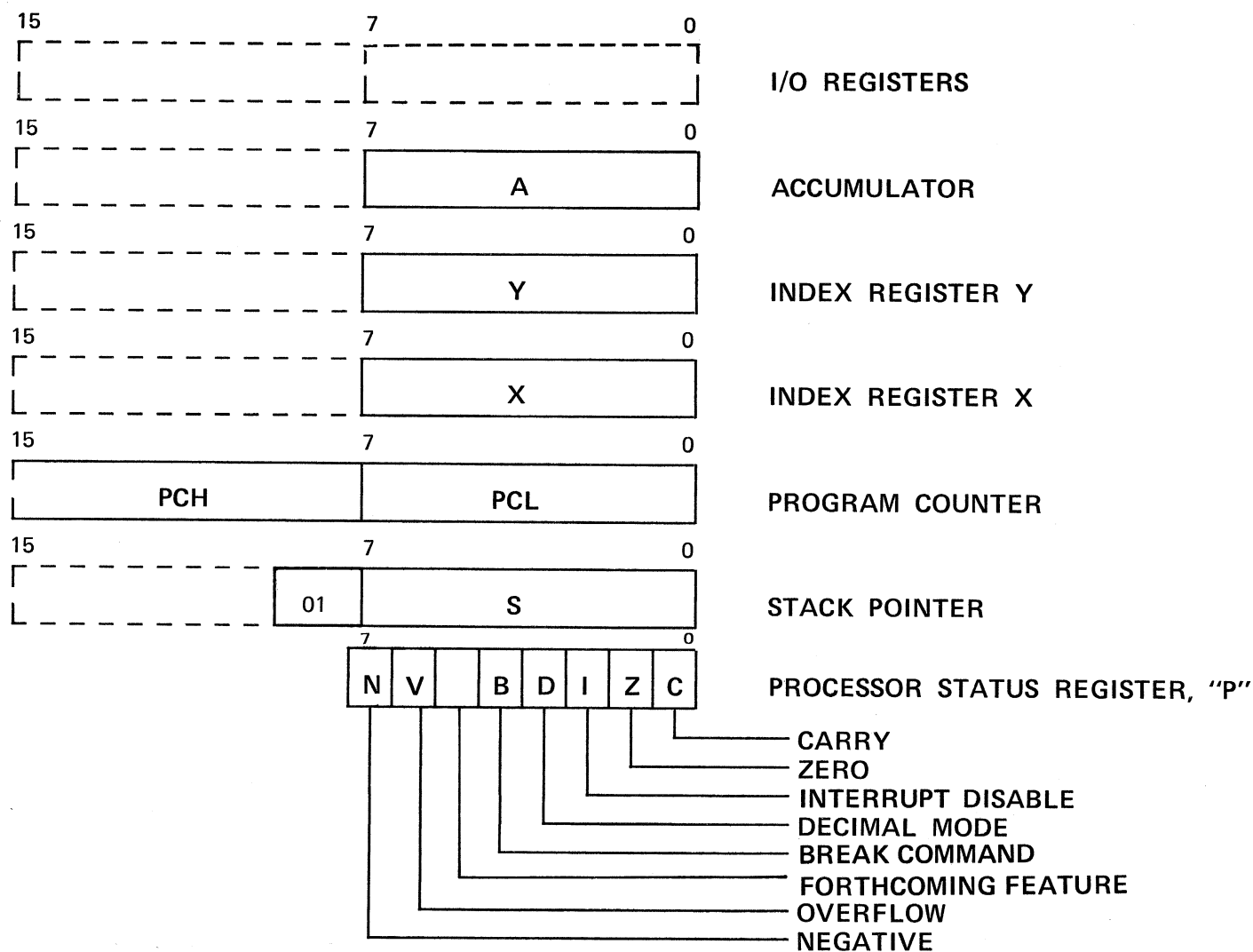
MCS650X-

the first Microprocessor family

FEATURES OF THE MCS650X

- Low Cost
- Eight bit parallel processing
- 55 Instructions
- Decimal and binary arithmetic
- Thirteen addressing modes
- True indexing capability
- Programmable stack pointer
- Variable length stack
- Interrupt capability
- Non-maskable interrupt
- Use with any type or speed memory
- Bi-directional Data Bus
- Instruction decoding and control
- Addressable memory range of up to 65K bytes
- "Ready" input
- Direct memory access capability
- Bus Compatible with MC6800
- On-the-Chip clock options
 - * External single clock input
 - * RC Time base input
 - * Crystal time base input
- 40 and 28 pin package versions
- Pipeline Architecture

PROGRAMMING MODEL MCS650X



* Solid line indicates currently available features
Dashed line indicates forthcoming members of family

POWERFUL ADDRESSING

Thirteen Addressing Modes including True Indexing

The MCS650X has thirteen modes of address. The first byte of each instruction is the operation code and specifies both the instruction and the addressing mode. The following table defines the addressing modes pertinent to each instruction with the corresponding execution time in clock cycles, where a 1MHz clock frequency implies a 1 microsecond cycle.

ACCUMULATOR ADDRESSING - This form of addressing is represented with a one byte instruction, implying an operation on the accumulator.

IMMEDIATE ADDRESSING - In immediate addressing, the operand is contained in the second byte of the instruction, with no further memory addressing required.

ABSOLUTE ADDRESSING - In absolute addressing, the second byte of the instruction specifies the eight low order bits of the effective address while the third byte specifies the eight high order bits. Thus, the absolute addressing mode allows access to the entire 65K bytes of addressable memory.

ZERO PAGE ADDRESSING - The zero page instructions allow for shorter code and execution times by only fetching the second byte of the instruction and assuming a zero high address byte. Careful use of the zero page can result in significant increase in code efficiency.

INDEXED ZERO PAGE ADDRESSING - (*X, Y indexing*) - This form of addressing is used in conjunction with the index register and is referred to as "Zero Page, X" or "Zero Page, Y". The effective address is calculated by adding the second byte to the contents of the index register. Since this is a form of "Zero Page" addressing, the content of the second byte references a location in page zero. Additionally due to the "Zero Page" addressing nature of this mode, no carry is added to the high order 8 bits of memory and crossing of page boundaries does not occur.

INDEXED ABSOLUTE ADDRESSING - (*X, Y, indexing*) This form of addressing is used in conjunction with X and Y index register and is referred to as "Absolute, X", and "Absolute, Y". The effective address is formed by adding the contents of X or Y to the address contained in the second and third bytes of the instruction. This mode allows the index register to contain the index or count value and the instruction to contain the base address. This type of indexing allows any location referencing and the index to modify multiple fields resulting in reduced coding and execution time.

IMPLIED ADDRESSING - In the implied addressing mode the address containing the operand is implicitly stated in the operation code of the instruction.

RELATIVE ADDRESSING - Relative addressing is used only with branch instructions and establishes a destination for the conditional branch.

The second byte of the instruction becomes the operand which is an "Offset" added to the contents of the lower eight bits of the program counter when the counter is set at the next instruction. The range of the offset is -128 to +127 bytes from the next instruction.

INDEXED INDIRECT ADDRESSING - In indexed indirect addressing (referred to as (Indirect, X)), the second byte of the instruction is added to the contents of the X index register, discarding the carry. The result of this addition points to a memory location on page zero whose contents is the low order eight bits of the effective address. The next memory location in page zero contains the high order eight bits of the effective address. Both memory locations specifying the high and low order bytes of the effective address must be in page zero.

INDIRECT INDEXED ADDRESSING - In indirect indexed addressing (referred to as (Indirect), Y), the second byte of the instruction points to a memory location in page zero. The contents of this memory location is added to the contents of the Y index register, the result being the low order eight bits of the effective address. The carry from this addition is added to the contents of the next page zero memory location, the result being the high order eight bits of the effective address.

ABSOLUTE INDIRECT - The second byte of the instruction contains the low order eight bits of a memory location. The high order eight bits of that memory location is contained in the third byte of the instruction. The contents of the fully specified memory location is the low order byte of the effective address. The next memory location contains the high order byte of the effective address which is loaded into the sixteen bits of the program counter.

INSTRUCTION ADDRESSING MODES AND RELATED EXECUTION TIMES (in clock cycles)

	ADC	AND	ASL	BCC	BCS	BEQ	BIT	BMI	BNE	BPL	BRK	BVC	BVS	CLC	CLD	CLI	CLV	CMP	CPX	CPY	DEC	DEX	DEY	EØR	INC	INX	INY	JMP
Accumulator	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Immediate	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Zero Page	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Zero Page, X	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Zero Page, Y	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Absolute	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Absolute, X	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Absolute, Y	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Implied	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Relative	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
(Indirect, X)	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*
(Indirect, Y)	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*
Absolute Indirect	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*	5*

* Add one cycle if indexing across page boundary

** Add one cycle if branch is taken, Add one additional if branching operation crosses page boundary

MCS6501 MICROPROCESSOR INSTRUCTION SET – ALPHABETIC SEQUENCE

*ADC	Add Memory to Accumulator with Carry	*JSR	Jump to New Location Saving Return Address
*AND	"AND" Memory with Accumulator	*LDA	Load Accumulator with Memory
*ASL	Shift Left One Bit (Memory or Accumulator)	*LDX	Load Index X with Memory
		LDY	Load Index Y with Memory
*BCC	Branch on Carry Clear	*LSR	Shift One Bit Right (Memory or Accumulator)
*BCS	Branch on Carry Set		
*BEQ	Branch on Result Zero	NOP	No Operation
*BIT	Test Bits in Memory with Accumulator	*ORA	"OR" Memory with Accumulator
*BMI	Branch on Result Minus	*PHA	Push Accumulator on Stack
*BNE	Branch on Result not Zero	PHP	Push Processor Status on Stack
*BPL	Branch on Result Plus	*PLA	Pull Accumulator from Stack
*BRK	Force Break	PLP	Pull Processor Status from Stack
*BVC	Branch on Overflow Clear	*ROL	Rotate One Bit Left (Memory or Accumulator)
*BVS	Branch on Overflow Set		
*CLC	Clear Carry Flag	*RTI	Return From Interrupt
CLD	Clear Decimal Mode	*RTS	Return From Subroutine
*CLI	Clear Interrupt Disable Bit	*SBC	Subtract Memory from Accumulator with Borrow
*CLV	Clear Overflow Flag	*SEC	Set Carry Flag
*CMP	Compare Memory and Accumulator	SED	Set Decimal Mode
*CPX	Compare Memory and Index X	*SEI	Set Interrupt Disable Status
CPY	Compare Memory and Index Y	*STA	Store Accumulator in Memory
*DEC	Decrement Memory by One	*STX	Store Index X in Memory
*DEX	Decrement Index X by One	STY	Store Index Y in Memory
DEY	Decrement Index Y by One	TAX	Transfer Accumulator to Index X
*EOR	"Exclusive-or" Memory with Accumulator	TAY	Transfer Accumulator to Index Y
*INC	Increment Memory by One	*TSX	Transfer Stack Pointer to Index X
*INX	Increment X by One	TXA	Transfer Index X to Accumulator
INY	Increment Y by One	*TXS	Transfer Index X to Stack Pointer
*JMP	Jump to New Location	TYA	Transfer Index Y to Accumulator

*Instructions similar to MC6800

HEADQUARTERS –

MOS TECHNOLOGY, INC. 950 Rittenhouse Road
Norristown, Pa. 19401, (215) 666-7950, TWX: 510/660/4033

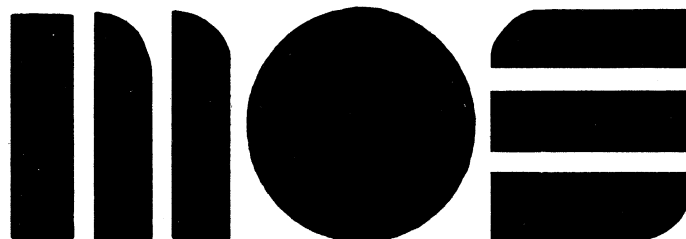
EASTERN REGION –

Mr. William Whitehead
MOS TECHNOLOGY, INC., Suite 312,
410 Jericho Turnpike, Jericho, N.Y. 11753
(516) 822-4240

WESTERN REGION –

MOS TECHNOLOGY, INC. 2172 Dupont Drive,
Patio Bldg., Suite 221. Newport Beach, CA. 92660
(714) 833-1600

Mr. Petr Sehnal, Regional Applications Mgr.
MOS TECHNOLOGY, INC., 26921 Grasmere Place,
Hayward, CA. 94542
(415) 881-8080



MOS TECHNOLOGY, INC.

**this document was
generously contributed by:
Barry Luokkala
Department of Physics,
Carnegie Mellon University**

scanned by:
www.commodore.international
2023-06-27